



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/599,086	06/21/2000	Eric S. Rockey	MS1-562US	6625
22801	7590	11/13/2006	EXAMINER	
LEE & HAYES PLLC			BONSHOCK, DENNIS G	
421 W RIVERSIDE AVENUE SUITE 500				
SPOKANE, WA 99201			ART UNIT	PAPER NUMBER
			2173	

DATE MAILED: 11/13/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<i>Office Action Summary</i>	Application No.	Applicant(s)
	09/599,086	ROCKEY ET AL.
Examiner	Art Unit	
Dennis G. Bonshock	2173	

Office Action Summary

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 01 September 2006.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 2-7,9-13,36-59 and 69-95 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 2-7,9-13,36-59 and 69-95 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 9-1-06.

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. ____ .
5) Notice of Informal Patent Application
6) Other: ____ .

NON-FINAL ACTION

Response to Amendment

1. It is hereby acknowledged that the following papers have been received and placed on record in the file: Response filed 9-1-2006.

Claims 1-95 have been examined.

Status of Claims:

2. Claims 2, 5-7, 9-11, 36-40, 43, 46-48, 50-54, 59, 80, 81, 84-89, and 92-95 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Patent # 5,572,648 and Samar, patent # 6,563,514, hereinafter Samar.
3. Claims 3, 4, 12, 13, 41, 42, 45, 49, 56, 57, 82, 83, 90, and 91 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Samar, and Meyer et al., Patent # 5,742,504, hereinafter Meyer.
4. Claim 58 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Samar, and Powers, III et al., Patent # 5,602,996, hereinafter Powers.
5. Claims 69-79 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baer et al., Patent # 6,611,840, hereinafter Baer, Meyer, Bibayan, and Samar.
6. Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Samar, and Baer.
7. Claim 55 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Samar, and Gayraud et al., Patent # 5,436,637, hereinafter Gayraud.
8. Claims 1, 8, 14-35, and 60-68 have been cancelled by the applicant.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 2, 5-7, 9-11, 36-40, 43, 46-48, 50-54, 59, 80, 81, 84-89, and 92-95 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Patent # 5,572,648 and Samar, patent # 6,563,514, hereinafter Samar.

12. With regard to claim 6, which teaches automatically displaying at least one command on a display for the user based on the user's context, Bibayan teaches, in column 4, lines 45-65, the dynamic tool palette display altering its size, shape, and contents in accordance with the context of the currently operating applet. Bibayan further teaches a tree based visibility expression in figure 5, and in column 5, lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the display. With regard to claim 6, further teaching determining a user's context comprises ascertaining a position of a user's cursor within a document provided by the application program, Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a

command, but doesn't specifically teach the position of the cursor alone determining the user context. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides and additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), but further teaches, in column 4, lines 39-45, the point of interest being determined by a pointer or cursor positioned by a user over the item of interest. Samar further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display, displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected element, cursor position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan and Samar before him at the time the invention was made to modify context sensitive display system of Bibayan to adapt based on a cursor position as did Samar. One would have been motivated to make such a combination because changing of the display based on the users cursor changes the item of focus without having to make a selection.

13. With regard to claim 2, which teaches automatically removing commands from the display responsive to a change in user context, Bibayan teaches, in column 5, line 45, that if tools are deemed not relevant then the display is modified to not include them.

14. With regard to claim 5, which teaches executing a command without requiring any action from a user other than selecting the command, Bibayan teaches, in column 4, line 53, an active toolbar where commands can be selected (executed) at any time.
15. With regard to claim 7, which teaches determining comprising ascertaining a user's selection within a document provided by the application program, Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command.
16. With regard to claim 9, which teaches the context pertaining to various tasks that the user may attempt to accomplish, Bibayan teaches, in column 5, line 2, the context modified with tool functions, which apply to the currently running folder.
17. With regard to claim 10, which teaches that the context pertains to one or more of: a type of document, and a state of a document, Bibayan teaches, in column 5, line 2, the context modified with tool functions which apply to the currently running folder.
18. With regard to claim 11, which teaches displaying being independent of a user selecting any displayed menu item, Bibayan teaches, in column 5, line 2, that modification is based on context changes.
19. With regard to claim 36, which teaches a method of exposing commands in a software application comprising; determining a user's context, Bibayan teaches, in column 1, lines 17-22, a display of a toolbar containing dynamic data that changes in accordance with the context of the environment or the currently operating application program. With regard to claim 36, which teaches determining comprising evaluating a portion of one or more expressions, expressions being associated with a context block, and defining a condition that describes one or more aspect of a user's interaction,

Bibayan teaches, in column 2, lines 41-51, defining a context environment which creates data for a designated container. The method contains steps of executing to determine environment (similar to evaluating expressions), and modify with respect to the context (the evaluation of the expressions). With regard to claim 36, further teaching automatically displaying at least one context block based on user context, Bibayan teaches, in column 5, line 2, modifying the toolbar when the context is modified. The context contains tool functions, which apply to the currently running folder. Bibayan further teaches a tree based visibility expression in figure 5, and in column 5, lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the display. Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command where a tree is evaluated to determine what items should be added or deleted, but doesn't specifically (or at least as clear as Samar) teach evaluation for each individual expression. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides and additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), but further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display, displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected element, cursor

position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan and Samar before him at the time the invention was made to modify the adaptive displayed elements of Bibayan, to be displayed selectively (on an individual basis), as did Samar. One would have been motivated to make such a combination because this provides a more adapted display.

20. With regard to claim 37, which teaches the expression evaluating to a Boolean value, Bibayan teaches, in column 4, line 66 through column 5, line 5, determining if the context modifies the toolbar or if it doesn't (much like a Boolean to modify? True or False).

21. With regard to claim 38, which teaches, that the context pertains to one or more of: a type of document, a state of a document, and a particular user's selection, Bibayan teaches, in column 5, line 2, the context modified with tool functions which apply to the currently running folder.

22. With regard to claim 39, which teaches displaying comprising displaying a context block having a title bar area that labels the context block, Bibayan teaches, in figure 4, a context block that has a given title associated with it.

23. With regard to claim 40, which teaches the title bar area configured to enable the context block to be toggled between expanded and collapsible states, Bibayan teaches, in figure 4 both the static and dynamic tool palettes being standard windows which have the functionality to be maximized, minimized, and closed.

24. With regard to claim 43, which teaches displaying a context block with a controls area that exposes the multiple commands to the user, Bibayan teaches, in column 5, line 2, determining whether to modify a tool display and displaying tool functions based on the context.

25. With regard to claim 46, which teaches a method of exposing commands in a software application program, comprising determining a users context, and displaying a dynamic toolbar based on the context, Bibayan teaches, in column 1, lines 17-22, a display of a toolbar containing dynamic data that changes in accordance with the context of the environment or the currently operating application program. With regard to claim 46, further teaching determining the user context being independent of the selection of a menu item, Bibayan teaches, in column 4, lines 45-65, the display being dependent on the context of an applet, not dependent on the user selecting any displayed menu item. With regard to claim 46, further teaching that while the commands are displayed, allowing the user to select and apply various commands multiple times, Bibayan teaches, in column 4, line 53, the dynamic tool palette being always active, where the operator can select any one of the commands. Bibayan further teaches a tree based visibility expression in figure 5, and in column 5, lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the display. Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command where a tree is evaluated to determine what items should be added or

deleted, but doesn't specifically (or at least as clear as Samar) teach evaluation for each individual expression. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides and additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), but further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display, displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected element, cursor position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan and Samar before him at the time the invention was made to modify the adaptive displayed elements of Bibayan, to be displayed selectively (on an individual basis), as did Samar. One would have been motivated to make such a combination because this provides a more adapted display.

26. With regard to claim 47, which teaches applying one or more of select commands, when selected by the user without further interaction, Bibayan teaches, in column 4, line 53, the dynamic tool palette being always active, where the operator can select any one of the commands.

27. With regard to claim 48, which teaches displaying comprising displaying the commands responsive to the user selecting from a menu bar supported by automatically-appearing and disappearing context block that contains context-sensitive

commands, Bibayan teaches, in column 5, lines 2-7 and lines 52-57, modifying the toolbar in response to a change in context, and a possible deletion of the toolbar.

28. With regard to claim 50, which teaches displaying comprises displaying the commands within a context pane having a title bar that labels the context pane and a controls area that exposes the commands to the user, Bibayan teaches, in column 5, line 2, and in figure 4, a dynamic toolbar which has separate title and tool sections.

29. With regard to claim 51, which teaches the context pane being non-collapsible, Bibayan teaches, in column 4, line 21, the user opening a context pane and a static pane by opening a specific application, therefor they are always displayed, the reference goes on the not specifically mention collapsing the context pane.

30. With regard to claim 52, which teaches that the context pane must be closed by the user, Bibayan teaches, in column 5, lines 52-57, that when all applets have been terminated the dynamic tool palette display can be terminated, as well.

31. With regard to claim 53, which teaches that the user requests the context pane be displayed, Bibayan teaches, in column 4, line 21, the user opens a context pane by opening a specific application.

32. With regard to claim 54, which teaches that some of the commands being context-sensitive and are disabled if they are out of context, Bibayan teaches, in column 5, line 45, that if tools are deemed not relevant then the display is modified to not include them.

33. With regard to claim 59, which teaches one or more computer readable instructions, which are executed by a computer, Bibayan teaches, in column 4, line 9, that the medium is a personal computer.

34. With regard to claim 80, which teaches, a method of exposing commands in a software application comprising: determining a user's context by ascertaining a position of a user's cursor within a document provided by the application program, Bibayan teaches, in column 4, lines 40-50, the user locating and initiating a command. With regard to claim 80, further teaching automatically displaying at least one command on a display for the user based on the user's context, Bibayan teaches, in column 4, lines 45-65, the dynamic tool palette display altering its size, shape, and contents in accordance with the context of the currently operating applet. Bibayan further teaches a tree based visibility expression in figure 5, and in column 5, lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the display. Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command where a tree is evaluated to determine what items should be added or deleted, but doesn't specifically (or at least as clear as Samar) teach evaluation for each individual expression. Samar teaches a system in which user's selection of a particular display cause context evaluation and further provides an additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), but further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display,

displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected element, cursor position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan and Samar before him at the time the invention was made to modify the adaptive displayed elements of Bibayan, to be displayed selectively (on and individual basis), as did Samar. One would have been motivated to make such a combination because this provides a more adapted display.

35. With regard to claim 81 and 89, which teach automatically removing commands from the display responsive to a change in user context, Bibayan teaches, in column 5, line 45, that if tools are deemed not relevant then the display is modified to not include them.

36. With regard to claims 84 and 92, which teach executing a command without requiring any action from a user other than selecting the command, Bibayan teaches, in column 4, line 53, an active toolbar where commands can be selected (executed) at any time.

37. With regard to claims 85 and 93, which teach the context pertaining to various tasks that the user may attempt to accomplish, Bibayan teaches, in column 5, line 2, the context modified with tool functions, which apply to the currently running folder.

38. With regard to claims 86 and 94, which teach that the context pertains to one or more of: a type of document, and a state of a document, Bibayan teaches, in column 5, line 2, the context modified with tool functions which apply to the currently running folder.

39. With regard to claim 87, which teaches displaying being independent of a user selecting any displayed menu item, Bibayan teaches, in column 5, line 2, that modification is based on context changes.

40. With regard to claim 88, which teaches, a method of exposing commands in a software application comprising: determining a user's context by ascertaining a position of a user's cursor within a document provided by the application program, Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command. With regard to claim 88, further teaching automatically displaying at least one command on a display for the user based on the user's context, Bibayan teaches, in column 4, lines 45-65, the dynamic tool palette display altering its size, shape, and contents in accordance with the context of the currently operating applet. With regard to claim 88, further teaching determining the user context being independent of the selection of a menu item, Bibayan teaches, in column 4, lines 45-65, the display being dependent on the context of an applet, not dependent on the user selecting any displayed menu item. Bibayan further teaches a tree based visibility expression in figure 5, and in column 5, lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the

display. Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command where a tree is evaluated to determine what items should be added or deleted, but doesn't specifically (or at least as clear as Samar) teach evaluation for each individual expression. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides and additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), but further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display, displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected element, cursor position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan and Samar before him at the time the invention was made to modify the adaptive displayed elements of Bibayan, to be displayed selectively (on an individual basis), as did Samar. One would have been motivated to make such a combination because this provides a more adapted display.

With regard to claim 95, which teach that the individual expressions are represented in a different tree structure, Sumar teaches, in column 5, lines 15-56, and in figure 4, different nodes of the tree with corresponding sub-nodes that have different sub tree structures. Furthermore, Figure 5, is a sub-decision tree for getting information from the server for display on the client.

41. Claims 3, 4, 12, 13, 41, 42, 45, 49, 56, 57, 82, 83, 90, and 91 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Samar, and Meyer et al., Patent # 5,742,504, hereinafter Meyer.

42. With regard to claims 3 and 4, Bibayan and Samar teach a system for displaying a dynamic toolbar for an application program (see Bibayan, column 1, line 17), but doesn't teach it being a document-centric program, that displaying does not obscure the document in which the user is working, and that the command is displayed in a modeless fashion in which the user can continue to work in the document while the command is displayed. Meyer teaches a system for displaying a dynamic toolbar similar to that of Bibayan and Samar, but further teaches a document-centric format, and that displaying does not obscure a document (see figure 3). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Meyer before him at the time the invention was made to modify the dynamic toolbar system of Bibayan and Samar to include the document-centric format and the display of this toolbar to be non-obtrusive, as did Meyer. One would have been motivated to make such a combination because having the ability to work with in a document while displaying all the toolbar options allows the user to use the toolbar options without obscuring the document.

43. With regard to claim 12, Bibayan teaches a system for determining a users context, displaying a dynamic toolbar for an application program (see column 1, line 17), automatically removing said command responsive to a change in context (see column

5, line 45), and the display being dependent on the context of an applet, not necessarily dependent on the user selecting any displayed menu item (see column 1, lines 24-30, column 4, lines 45-65, and column 5, lines 19-32), but doesn't teach that the command is displayed in a modeless fashion in which the user can continue to work in the document while the command is displayed. Meyer teaches a system for displaying a dynamic toolbar similar to that of Bibayan, but further teaches the toolbar being displayed in a non-intrusive manner (see figure 3). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan and Meyer before him at the time the invention was made to modify the dynamic toolbar system of Bibayan to include the display of this toolbar to be non-intrusive, as did Meyer. One would have been motivated to make such a combination because having the ability to work within a document while displaying all the toolbar options allows the user to use the toolbar options without obscuring the document. Bibayan further teaches a tree based visibility expression in figure 5, and in column 5, lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the display. Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command where a tree is evaluated to determine what items should be added or deleted, but doesn't specifically (or at least as clear as Samar) teach evaluation for each individual expression. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides and additional display to a user (see column 4, lines 39-

45, column 9, lines 30-34, and figures 6a and 6b), but further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display, displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected element, cursor position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Meyer, and Samar before him at the time the invention was made to modify the adaptive displayed elements of Bibayan and Meyer, to be displayed selectively (on an individual basis), as did Samar. One would have been motivated to make such a combination because this provides a more adapted display.

44. With regard to claim 13, which teaches determining comprises ascertaining a position of a user's cursor with in a document provided by the application program and ascertaining a user's selection within a document provided by the application program Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command, but doesn't specifically teach the position of the cursor alone determining the user context. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides and additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), but further teaches, in column 4, lines 39-45, the point of interest being determined by a pointer or cursor positioned by a user over the item of interest. It would have been obvious to one of

ordinary skill in the art, having the teachings of Bibayan, Meyer, and Samar before him at the time the invention was made to modify context sensitive display system of Bibayan and Meyer to adapt based on a cursor position as did Samar. One would have been motivated to make such a combination because changing of the display based on the users cursor changes the item of focus without having to make a selection.

45. With regard to claim 41, Bibayan teaches a system for determining a users context, displaying a dynamic toolbar for an application program (see column 1, line 17), but doesn't teach a title bar area containing a display button configured to enable a menu associated with the context block. Meyer teaches a system for displaying a dynamic toolbar similar to that of Bibayan, but further teaches the toolbar containing a menu that is associated with a context block. It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Meyer before him at the time the invention was made to modify dynamic toolbar system of Bibayan and Samar to include the display of this menu bar, as did Meyer. One would have been motivated to make such a combination because the use of a menu bar in a toolbar can add a sub-menu with added functionality.

46. With regard to claim 42, which teaches a menu button that contains links to one or more context panes, each comprising additional context sensitive information, Meyer further teaches, in column 4, line 64 through column 5, line 16, menu buttons that open additional context-sensitive panes.

47. With regard to claim 45, Bibayan teaches a system for determining a users context, displaying a dynamic toolbar for an application program (see column 1, line 17),

and for automatically removing said command responsive to a change in context (see column 5, line 45), but doesn't teach that the command is displayed in a modeless fashion in which the user can continue to work in the document while the command is displayed. Meyer teaches a system for displaying a dynamic toolbar similar to that of Bibayan, but further teaches the toolbar being displayed in a non-intrusive manner (see figure 3). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Meyer before him at the time the invention was made to modify the dynamic toolbar system of Bibayan and Samar to include the display of this toolbar to be non-intrusive, as did Meyer. One would have been motivated to make such a combination because having the ability to work with in a document while displaying all the toolbar options allows the user to use the toolbar options without obscuring the document.

48. With regard to claim 49, Bibayan teaches, in column 1, lines 17-22, a display of a toolbar containing dynamically data in accordance with the context of the environment or the currently operating application program; however, he fails to teach displaying the commands in a modeless manner. Meyer teaches a system for displaying a dynamic toolbar similar to that of Bibayan and Samar, but further teaches displaying commands in a modeless manner (see figure 3). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Meyer before him at the time the invention was made to modify the Dynamic toolbar system of Bibayan and Samar to include the display of this toolbar to be non-obtrusive, as did Meyer. One would have been motivated to make such a combination because having the ability to work with in a

document while displaying all the toolbar options allows the user to use the toolbar options without obscuring the document.

49. With regard to claims 56 and 57, Bibayan teaches, in column 1, lines 17-22, a display of a toolbar containing dynamically data in accordance with the context of the environment or the currently operating application program, however, he fails to teach a help feature accessible via and icon on the title bar and this help being displayed in a modeless fashion. Meyer teaches a system for displaying a dynamic toolbar similar to that of Bibayan, but further teaches displaying a help command in a modeless fashion (see column 21, line 65 and figure 3). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Meyer before him at the time the invention was made to modify the dynamic toolbar system of Bibayan and Samar to include the display of a help icon in the toolbar in a modeless fashion. One would have been motivated to make such a combination because having the ability to access a help command without moving or minimizing screens has a time saving benefit.

50. With regard to claims 82, 83, 90, and 91, Bibayan teaches a system for displaying a dynamic toolbar for an application program (see column 1, line 17), but doesn't teach it being a document-centric program, that displaying does not obscure the document in which the user is working, and that the command is displayed in a modeless fashion in which the user can continue to work in the document while the command is displayed. Meyer teaches a system for displaying a dynamic toolbar similar to that of Bibayan and Samar, but further teaches a document-centric format,

and that displaying does not obscure a document (see figure 3). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Meyer before him at the time the invention was made to modify the dynamic toolbar system of Bibayan and Samar to include the document-centric format and the display of this toolbar to be non-obtrusive, as did Meyer. One would have been motivated to make such a combination because having the ability to work with in a document while displaying all the toolbar options allows the user to use the toolbar options without obscuring the document.

51. Claim 58 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Samar, and Powers, III et al., Patent # 5,602,996, hereinafter Powers.

52. With regard to claim 58, Bibayan teaches, in column 1, lines 17-22, a display of a toolbar containing dynamic data in accordance with the context of the environment or the currently operating application program; however, he fails to teach multiple context panes stackable in a queue. Powers teaches a method of handling panes similar to that of Bibayan and Samar, but further teaches, in column 3, lines 9-35, overlapping windows that can contain button functions, layering on top of one another in which only the top window of the stack is active. It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Powers before him at the time the invention was made to modify the display of toolbars containing dynamic data of Bibayan and Samar to contain the hierarchical organization as did Powers. One would have been motivated to make such a combination because ordering of window is

important to make sure that secondary functions of a previous context are implemented before moving into a different context.

53. Claims 69-79 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baer et al., Patent # 6,611,840, hereinafter Baer, Meyer, Bibayan, and Samar.

54. With regard to claim 69, Baer teaches a computer system comprising a single application program comprising a single navigable window, further teaching multiple different functionalities to which the single navigable window can be navigated by a user (see figures 8-21, column 2, lines 14-21, column 45, lines 45-50, and column 49, lines 13-22), but doesn't teach a context sensitive command area for navigating the windows or a single application program configured to automatically change command sets that are presented to the user, or that commands are displayed independent of the user selecting a displayed menu item. Meyer teaches a dynamic windowing environment similar to that of Baer but further teaches, in column 21, line 65 through column 22, line 11, a context sensitive toolbar with volatile elements that only appear in specific context, and in column 21, line 65 through column 22, line 60, that volatile buttons, that appear only in a specific context, are linked to a specific window or sub window that is visible and nowhere states that it is dependent on a selection of a menu item. It would have been obvious to one of ordinary skill in the art, having the teachings of Baer and Meyer before him at the time the invention was made to modify the windowing environment of Baer to include the dynamic command areas as did Meyer, and the displayed commands being dependent on the specific context of the window (see column 21, line

65 through column 22, line 11). One would have been motivated to make such a combination because dynamic command areas provide useful functions that can be preformed on the live document, while omitting non-applicable tools. Baer and Meyer, however don't specifically teach a tree based visibility expression that defines conditions of interaction. Bibayan teaches a system in which an adaptive display is provided to a user to supply context sensitive material to the user (see column 1, lines 50-63, column 2, lines 12-22, and column 4, lines 45-55), similar to that of Baer and Meyer, but further teaches a tree based visibility expression in figure 5, and in column 5, lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the display. Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command where a tree is evaluated to determine what items should be added or deleted, but doesn't specifically (or at least as clear as Samar) teach evaluation for each individual expression. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides an additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), similar to that of Baer, Meyer, and Bibayan, but further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display, displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected

element, cursor position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary skill in the art, having the teachings of Baer, Meyer, Bibayan, and Samar before him at the time the invention was made to modify the adaptive displayed elements of Baer, Meyer, and Bibayan, to be displayed selectively (on an individual basis), as did Samar. One would have been motivated to make such a combination because this provides a more adapted display.

55. With regard to claim 70, which teaches the single application program configured to provide navigation instrumentalities associated with the window, and these functionalities being configured for use, Baer further teaches in figures 8-21, navigating between different functionalities through the use of navigation buttons such as HOME, REGISTER LOGIN, LIBRARY, and HELP.

56. With regard to claim 71, which teaches that the navigation instrumentalities comprises links associated with each different functionalities, Baer further teaches in figures 8-21, navigating between different functionalities through the use of navigation buttons which contain links to different pages such as HOME, REGISTER LOGIN, LIBRARY, and HELP.

57. With regard to claim 72, which teaches one of the navigation instrumentalities comprising browser-like navigation buttons to navigate a window between different functionalities, Baer teaches, in figures 8-21 and in column 2, lines 14-21, a web based browser system, having embedded links to related pages.

58. With regard to claim 73, with teaches multiple different functionalities comprising document-centric functionalities, Baer teaches, in figures 8-21 and in column 2, lines 14-21, a web based browser system, which is document centered.

59. With regard to claim 74, Baer teaches a computer system comprising a single application program, further teaching multiple different functionalities to which the single navigable window can be navigated by a user (see figures 8-21, column 2, lines 14-21, column 45, lines 45-50, and column 49, lines 13-22) and incorporating different functionalities in an extensible manner so that the user can use the single navigable window to navigate to the different functionalities (see figures 8-21 column 45, lines 45-50, and column 49, lines 13-22), but doesn't teach a context sensitive command area for navigating the windows or a single application program configured to automatically change command sets that are presented to the user, or a toolbar that is context sensitive and not dependent on a menu selection. Meyer teaches a dynamic windowing environment similar to that of Baer but further teaches, in column 21, line 65 through column 22, line 11, a context sensitive toolbar with volatile elements that only appear in specific context, and the displayed commands being dependent on the specific context of the window, not a menu selection (see column 21, line 65 through column 22, line 11). It would have been obvious to one of ordinary skill in the art, having the teachings of Baer and Meyer before him at the time the invention was made to modify the windowing environment of Baer to include the dynamic command areas as did Meyer. One would have been motivated to make such a combination because dynamic command areas provide useful functions that can be preformed on the live document,

while omitting none applicable tools. Baer and Meyer, however don't specifically teach a tree based visibility expression that defines conditions of interaction. Bibayan teaches a system in which a adaptive display is provided to a user to supply context sensitive material to the user (see column 1, lines 50-63, column 2, lines 12-22, and column 4, lines 45-55), similar to that of Baer and Meyer, but further teaches a tree based visibility expression in figure 5, and in column 5, lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the display. Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command where a tree is evaluated to determine what items should be added or deleted, but doesn't specifically (or at least as clear as Samar) teach evaluation for each individual expression. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides and additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), similar to that of Baer, Meyer, and Bibayan, but further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display, displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected element, cursor position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary

skill in the art, having the teachings of Baer, Meyer, Bibayan, and Samar before him at the time the invention was made to modify the adaptive displayed elements of Baer, Meyer, and Bibayan, to be displayed selectively (on an individual basis), as did Samar. One would have been motivated to make such a combination because this provides a more adapted display.

60. With regard to claim 75, which teaches the single application program configured to provide navigation instrumentalities associated with the window, and these functionalities being configured for use, Baer further teaches in figures 8-21, navigating between different functionalities through the use of navigation buttons such as HOME, REGISTER LOGIN, LIBRARY, and HELP.

61. With regard to claim 76, which teaches that the navigation instrumentalities comprise links associated with each different functionalities, Baer further teaches in figures 8-21, navigating between different functionalities through the use of navigation buttons which contain links to different pages such as HOME, REGISTER LOGIN, LIBRARY, and HELP.

62. With regard to claim 77, which teaches one of the navigation instrumentalities comprising browser-like navigation buttons to navigate a window between different functionalities, Baer teaches, in figures 8-21 and in column 2, lines 14-21, a web based browser system, having embedded links to related pages.

63. With regard to claim 78, Baer teaches a computer method, further teaching multiple different functionalities to which the single navigable window can be navigated by a user (see figures 8-21, column 2, lines 14-21, column 45, lines 45-50, and column

49, lines 13-22), receiving user input to indicate a selection of a particular functionality (see column 45, line 51), and incorporating different functionalities in an extensible manner so that the user can use the single navigable window to navigate to the different functionalities (see figures 8-21 column 45, lines 45-50, and column 49, lines 13-22), but doesn't teach a context sensitive command area for navigating the windows or a single application program configured to automatically change command sets that are presented to the user, or that commands are displayed independent of the user selecting a displayed menu item. Meyer teaches a dynamic windowing environment similar to that of Baer but further teaches, in column 21, line 65 through column 22, line 11, a context sensitive toolbar with volatile elements that only appear in specific context, and the displayed commands being dependent on the specific context of the window (see column 21, line 65 through column 22, line 11). It would have been obvious to one of ordinary skill in the art, having the teachings of Baer and Meyer before him at the time the invention was made to modify the windowing environment of Baer to include the dynamic command areas as did Meyer. One would have been motivated to make such a combination because dynamic command areas provide useful functions that can be preformed on the live document, while omitting none applicable tools. Baer and Meyer, however don't specifically teach a tree based visibility expression that defines conditions of interaction. Bibayan teaches a system in which a adaptive display is provided to a user to supply context sensitive material to the user (see column 1, lines 50-63, column 2, lines 12-22, and column 4, lines 45-55), similar to that of Baer and Meyer, but further teaches a tree based visibility expression in figure 5, and in column 5,

lines 19-56, where the addition and deletion of elements to the display is effected by user interaction with the system, and the corresponding route being followed in the tree to determine if a particular element should be added to the display or removed from the display. Bibayan teaches, in column 4, lines 40-44, the user locating and initiating a command where a tree is evaluated to determine what items should be added or deleted, but doesn't specifically (or at least as clear as Samar) teach evaluation for each individual expression. Samar teaches a system in which users selection of a particular display cause context evaluation and further provides and additional display to a user (see column 4, lines 39-45, column 9, lines 30-34, and figures 6a and 6b), similar to that of Baer, Meyer, and Bibayan, but further teaches a further evaluation of a tree structure for each particular item a user hovers over on the display, displaying the particular item should the user hover over it for a specified period of time (see figure 4, and column 5, lines 15-56). Each of the states between user selection of a GUI element (start) and the providing a user with a customized GUI element (end) based on (elapsed time, selected element, cursor position, etc), is represented by an expression in a decision tree (in both Bibayan [figure 5] and Samar [figures 4 and 5]). It would have been obvious to one of ordinary skill in the art, having the teachings of Baer, Meyer, Bibayan, and Samar before him at the time the invention was made to modify the adaptive displayed elements of Baer, Meyer, and Bibayan, to be displayed selectively (on and individual basis), as did Samar. One would have been motivated to make such a combination because this provides a more adapted display.

64. With regard to claim 79, which teaches automatically removing said a least one command from the display responsive to change in the user's context, Mayer further teaches, in column 22, line 5, volatile buttons appearing only in specific contexts.

65. Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Samar, and Baer.

66. With regard to claim 44, Bibayan and Samar teach a method of displaying dynamic toolbars responsive to user context, but doesn't teach a command display being defined in HTML. Baer teaches a windowing environment similar to that of Bibayan and Samar, but further teaches, the system being Internet based (HTML). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Baer before him at the time the invention was made to modify the dynamic toolbars of Bibayan to be used in an Internet environment. One would have been motivated to make such a combination because use of the dynamic toolbar in a browser will allow for easier navigation through the Internet.

67. Claim 55 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bibayan, Samar, and Gayraud et al., Patent # 5,436,637, hereinafter Gayraud.

68. With regard to claim 55, Bibayan and Samar teach a method of displaying dynamic toolbars responsive to user context, but doesn't teach context panes including context sensitive information. Gayraud teaches a windowing environment similar to that of Bibayan and Samar, but further teaches, a help feature for toolbars in which info is

displayed when a users mouse moves over an icon (see column 3, lines 45-55). It would have been obvious to one of ordinary skill in the art, having the teachings of Bibayan, Samar, and Gayraud before him at the time the invention was made to modify the method of displaying dynamic toolbars of Bibayan and Samar to include the help feature of Gayraud. One would have been motivated to make such a combination because hints for tools can help a user to be able to make sense of a complex set of toolbar icons.

Response to Arguments

69. The arguments filed on 9-1-2006 have been fully considered but they are not persuasive. Reasons set forth below.

70. The Applicants' argue that neither Bibayan nor Samar teach that "individual expressions are represented in a tree data structure".

71. In response, the Examiner respectfully submits that Bibayan (figure 5) and Samar (figures 4 and 5) both teach Decision Trees (as defined in Microsoft Computer Dictionary, Fifth Edition, page 149), for use in creating links between expressions representing system states. This tree of states is used for evaluating whether or not to present custom information based upon selection of a GUI object.

Conclusion

72. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis G. Bonshock whose telephone number is (571) 272-4047. The examiner can normally be reached on Monday - Friday, 6:30 a.m. - 4:00 p.m.

73. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine Kincaid can be reached on (571) 272-4063. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

74. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

11-6-06
dgb



RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2173